

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problems Mailbox.**

**THIS PAGE BLANK (USPTO)**

①9 BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENTAMT

⑫ Übersetzung der  
europäischen Patentschrift

⑧7 EP 0 379 998 B1

⑩ DE 690 30 772 T 2

⑤1 Int. Cl.<sup>6</sup>:  
G 06 F 7/52

②1 Deutsches Aktenzeichen: 690 30 772.1  
⑥6 Europäisches Aktenzeichen: 90 101 112.2  
⑥6 Europäischer Anmeldetag: 19. 1. 90  
⑧7 Erstveröffentlichung durch das EPA: 1. 8. 90  
⑧7 Veröffentlichungstag  
der Patenterteilung beim EPA: 28. 5. 97  
④7 Veröffentlichungstag im Patentblatt: 8. 1. 98

③0 Unionspriorität:

13191/89 24.01.89 JP

⑦3 Patentinhaber:

Oki Electric Industry Co., Ltd., Tokio/Tokyo, JP

⑦4 Vertreter:

Betten & Resch, 80469 München

⑧4 Benannte Vertragsstaaten:

DE, FR, GB

⑦2 Erfinder:

Kihara, Kouichi, c/o Oki Electric Industry Co. Ltd.,  
Minato-ku, Tokyo, JP; Yamamoto, Kazushige, Oki  
Electric Industry Co. Ltd., Minato-ku, Tokyo, JP

⑤4 Dividierer zur hochgeschwindigen Ausführung einer arithmetischen Operation

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patentamt inhaltlich nicht geprüft.

DE 690 30 772 T 2

DE 690 30 772 T 2

690 30 772.1-08

HINTERGRUND DER ERFINDUNGGEBIET DER ERFINDUNG

Die vorliegende Erfindung bezieht sich im allgemeinen auf numerische arithmetische Verarbeitungseinrichtungen und insbesondere auf einen für Microcomputer passenden Dividierer.

BESCHREIBUNG DES STANDS DER TECHNIK

Herkömmliche Dividierer verwenden ein sogenanntes "Non-Restoring"-Verfahren, d.h. ein Verfahren ohne Bildung eines positiven Restes, um binäre numerische Daten zu dividieren. Die Dividierer haben in der Regel eine Vielzahl von Registern zum Speichern eines Divisors, eines Dividenden, eines Quotienten und anderer ähnlicher Werte, sowie eine Schiebereinrichtung für arithmetisches Verschieben.

Im Non-Restoring-Verfahren wird ein Quotient durch Subtrahieren eines Divisors von einem Dividenden und durch sequentielles Wiederholen einer arithmetischen Operation wie einer Berechnungsverschiebung erzeugt. Angenommen, es werden numerische Daten in binärer Schreibweise dividiert, so wird eine Binärzahl in einem numerischen Format dargestellt mit einem Festkomma und einem Zweierkomplement, wie in Fig. 3 dargestellt ist. Dann wird ein Quotient Q durch das Dividieren eines Dividenden A durch einen Divisor B wie folgt erzeugt:

$$Q = A / B \quad \dots (1)$$

Wenn

$$0 \leq A, 0 < B, A < 2B \quad \dots (2)$$

dann

$$0 \leq Q < 2 \quad \dots (3)$$

Das numerische Format des Quotienten Q, wie in Fig. 3 gezeigt, ordnet ein Bit einer Vorzeichenposition, ein Bit einem ganzzahligen Teil, und N Bits den Stellen hinter dem Komma zu.

Ein Verfahren zum Bestimmen des Quotienten Q wie in der Darstellung der Gleichung (1) wird unter Bezugnahme auf Fig. 4 beschrieben. Angenommen, ein binärer Dividierer dividiert einen Dividenten A1 durch einen Divisor B1. Dann beginnt das in Fig. 4 dargestellte Verfahren mit einem Schritt 41, um ein Register Q zu laden, das einem Quotienten mit einem Anfangswert, der gleich logischen NULLen ist, zugeordnet ist. Der Divident A1 bzw. der Divisor B1 wird in die Register A bzw. B geladen. Der Divisor B1 wird vom Dividenten A1 subtrahiert, und das Subtraktionsergebnis C1 wird in einem Register C gespeichert (Schritt 42). Wenn das Subtraktionsergebnis C1 positiv ist, wie im Schritt 43 bestimmt wird, wird eine (logische) EINS im Register Q gespeichert und arithmetisch um ein Bit nach links verschoben. Der resultierende Wert im Register Q soll dann C2 sein (Schritt 44). Zur selben Zeit wird ein Wert C3, der durch Verdopplung des Ergebnisses C1 erzeugt wurde, in das Register A geschrieben (Schritt 45).

Wenn andererseits das Subtraktionsergebnis C1 negativ ist, wird eine NULL im Register Q gespeichert und arithmetisch um ein Bit nach links verschoben (Schritt 46). Der so im Register Q gespeicherte Wert soll C4 genannt werden. Des weiteren wird ein Wert A2, der durch das Verdoppeln des Wertes A1 erzeugt wurde, in das Register A geschrieben (Schritt 47).

Die oben beschriebene Abfolge der Schritte (Schritte 42 bis 47) wird N+1 mal wiederholt, wobei N die Anzahl der Bits ist, die den Stellen hinter dem Komma zugewiesen sind (Schleife 48). Folglich wird ein Quotient Q im Register C gespeichert.

Ein herkömmlicher Mikroprozessor hat nicht die exklusive Hardware für die Durchführung der oben besprochenen Division. Solch ein Mikroprozessor hat einen

Nachteil, daß es einige zig Maschinenzyklen pro Bit eines Quotienten dauert, so daß es schwer ist, eine Division mit hoher Geschwindigkeit durchzuführen.

Aus EP-A2-0258051 ist ein digitaler Signalprozessor mit einer Divisionsfunktion bekannt. Dieser digitale Singalprozessor für die Division einer positiven Zahl in  $N+1$  Prozessorzyklen durch, wobei  $N$  gleich der Anzahl der Bits im Dividenten ist. Dies wird dadurch erreicht, daß eine arithmetische Logikeinheit im Prozessor verwendet wird, die in zwei selektiv verkettete, unabhängig voneinander steuerbare Abschnitte unterteilt ist, so daß die Werte darin selektiv verarbeitet werden können gesteuert durch einen einzigen Algorithmus.

Ein Divident wird Bit um Bit in der Richtung von einem höchstwertigen Bit (MSB) zu einem niedrigstwertigen Bit (LSB) in einen Akkumulator eingegeben. Das Vorzeichen des Operationsergebnisses wird dann verwendet, um zu bestimmen, ob im nächsten Schritt eine Addition oder eine Subtraktion des Divisors durchgeführt wird.

### ZUSAMMENFASSUNG DER ERFINDUNG

Es ist daher eine Aufgabe der vorliegenden Erfindung, einen Dividierer zu liefern, der eine einfache Schaltungskonstruktion aufweist und eine Binärzahl-Divisionsverarbeitung in einer kurzen Zeit durchführen kann.

Die vorliegende Erfindung ist in Anspruch 1 offenbart. Die Ansprüche 2 bis 6 offenbaren weitere Ausführungsbeispiele der Erfindung.

Ein Dividierer zur Erzeugen eines Quotienten durch das Dividieren eines Dividenten durch einen Divisor enthält eine erste Halteschaltung zum Halten der für den Dividenten kennzeichnenden Divident-Daten und eine zweite Halteschaltung zum Halten der für den Divisor kennzeichnenden Divisor-Daten. Eine Operationseinrichtung erzeugt entweder eine Summe oder eine Differenz zwischen den in der ersten Halteeinrichtung gehaltenen Divident-Daten und den in der zweiten Halteeinrichtung gehaltenen Divisor-Daten. Eine dritte Halteeinrichtung hält

Vorzeichenbit-Daten, die in den Ergebnisdaten enthalten sind, welche für ein von der Operationseinrichtung erzeugtes Operationsergebnis kennzeichnend sind. Eine Invertierschaltung invertiert die Vorzeichenbit-Daten, um invertierte Vorzeichenbit-Daten zu erzeugen. Eine erste Schiebeschaltung verschiebt sequentiell die invertierten Vorzeichenbit-Daten von der niedrigstwertigen Bitposition (LSB-Position) immer dann, wenn die invertierten Vorzeichenbit-Daten aus der Invertierschaltung eingegeben werden, während die invertierten Vorzeichenbit-Daten gehalten werden. Eine zweite Schiebeschaltung verschiebt arithmetisch die von der Operationseinrichtung erzeugten Ergebnisdaten um ein Bit nach links, während eine logische NULL in der LSB-Position gespeichert wird, und liefert die Ergebnisdaten an die erste Halteschaltung. Eine Steuerungseinrichtung steuert die Operationseinrichtung sowie die erste und die zweite Verschiebeschaltung, so daß die iterative Divisionsverarbeitung wiederholt ausgeführt wird.

#### KURZE BESCHREIBUNG DER ZEICHNUNGEN

Die Aufgaben und Merkmale der vorliegenden Erfindung werden deutlicher bei der Betrachtung der nachfolgenden detaillierten Beschreibung zusammen mit den beigefügten Zeichnungen, in denen

Fig. 1 ein schematisches Blockdiagramm ist, das eine Ausführungsform des erfindungsgemäßen Dividierers zeigt;

Fig. 2 ein Flußdiagramm ist, in dem eine spezifische Operation des in Fig. 1 gezeigten Dividierers dargestellt wird;

Fig. 3 ein spezifisches numerisches Format zeigt, das für den in Fig. 1 gezeigten Dividierer verwendet werden kann; und

Fig. 4 ein Flußdiagramm ist, das für einen Arbeitsablauf, der für einen Dividierer nach Stand der Technik typisch ist, kennzeichnend ist.

#### BESCHREIBUNG DER BEVORZUGTEN AUSFÜHRUNGSFORM

Unter Bezugnahme auf Fig. 1 der Zeichnungen wird ein erfindungsgemäßer Dividierer beispielsweise in einen Mikroprozessor eingebaut. Die veranschaulichen-

de Ausführungsform dividiert einen binären Dividenten A durch einen Divisor B ( $0 < A$ ,  $0 < B$ ,  $A < 2B$ ). Ein binäres numerisches Datenformat wird aus Zweierkomplementen und einem Festkomma gebildet, wie in Fig. 3 gezeigt ist. Folglich gilt:

$$0 \leq Q < 2$$

In Fig. 3 weist das numerische Format ein Bit für eine Vorzeichenposition, ein Bit für einen ganzzahligen Teil und N Bits für die Stellen hinter dem Komma auf.

Der in Fig. 1 gezeigte Dividierer umfaßt einen Addierer/Subtrahierer oder arithmetische Einheit (arithmetic unit, AU) 3, ein B-Register 2 und ein A-Register 1. Das B-Register 2 ist mit einem Eingangsanschluß des Addierers/Subtrahierers 3 verbunden, während das A-Register 1 mit einem anderen Eingangsanschluß des Addierers/Subtrahierers 3 und mit einem Ausgang des Addierers/Subtrahierers 3 über eine Schiebereinrichtung (SFT) 7 verbunden ist. Der Dividierer weist des weiteren ein Invertierglied 5, ein D-Flipflop (DFF) 4, einen Schleifenzähler (LC) 8 sowie ein Schieberegister (Q sfr) 6 auf.

Das A-Register 1 und das B-Register 2 dienen als Schaltung zum Speichern eines Dividenten bzw. eines Divisors. Das D-Flipflop 4 hält ein Vorzeichenbit (SGN) 9 eines Operationsergebnisses vom Addierer/Subtrahierer 3 einen Maschinenzklus lang. Das Invertierglied 5 invertiert das Vorzeichenbit 9 der Ausgabe des Addierers/Subtrahierers 3 und liefert das invertierte Vorzeichenbit an das Schieberegister 6.

Der Schleifenzähler 8 wird initialisiert auf die wiederholte Anzahl  $N+1$ , die für die in Fig. 2 gezeigte Divisionsoperation 27 benötigt wird. Der Schleifenzähler 8 wird jedes Mal, wenn die iterative Verarbeitung beendet ist, dekrementiert und ist am Ende bis auf Null dekrementiert.

Während der sequentiellen Dekrementierung bis auf 0 (Null) während der Division hält der Schleifenzähler 8 ein Steuersignal 10 auf einem aktiven Pegel, um den Addierer/Subtrahierer 3, das Schieberegister 6, die Schiebereinrichtung 7 und



den Programmzähler 11, darüber zu informieren, daß die Divisionsverarbeitung abläuft.

Nur wenn die Divisionsverarbeitung abläuft, verschiebt das Schieberegister 6 sequentiell die Ausgabe 60 des Invertierglieds 5 nach links von der niedrigsten oder niedrigstwertigen Bitposition (LSB-Position), bis es den gesamten Quotienten  $Q$  speichert. Die Schiebereinrichtung 7 speichert die Ausgabe des Addierers/Subtrahierers 3 nur dann, wenn die Divisionsverarbeitung ausgeführt wird; die numerischen Daten werden arithmetisch um ein Bit nach links in der Schiebereinrichtung 7 verschoben, wobei eine NULL in der LSB-Position gespeichert wird. Die Ergebnisdaten werden von der Schiebereinrichtung 7 an das A-Register 1 geliefert.

Der Addierer/Subtrahierer 3 ist als eine arithmetische Operationsschaltung ausgeführt, um eine Addition oder Subtraktion mit den Ausgaben des A-Registers 1 und des B-Registers 2 durchführen zu können. Ob die arithmetische Operationseinheit 3 eine Addition oder eine Subtraktion durchführen soll, wird bestimmt durch die Referenzierung des Ausgangs 62 des D-Flipflops 4, nur wenn die Divisionsverarbeitung ausgeführt wird. Genauer gesagt werden Subtraktion bzw. Addition ausgewählt, wenn die Ausgabe 62 des Flipflops 4 eine NULL bzw. eine EINS ist.

Ein Divisionsverfahren, das für die veranschaulichende Ausführungsform kennzeichnend ist, weist einen einzigen Algorithmus wie folgt auf. Der in der Gleichung (1) angegebene Quotient  $Q$  wird von einer spezifischen Abfolge von Schritten, wie sie in Fig. 2 dargestellt ist, erzeugt.

In Fig. 2 werden ein Dividend  $A$  und ein Divisor  $B$  vorab im A-Register 1 bzw. im B-Register 2 gespeichert. Der Schleifenzähler 8 wird anfangs mit der Anzahl  $N+1$  geladen, die angibt, wieviele Male die iterative Divisionsoperation durchgeführt werden soll. Das Schieberegister 6 und das D-Flipflop 4 werden zunächst zurückgesetzt, d.h. eine NULL wird in allen Bits gespeichert.

Nach der oben beschriebenen Initialisierung wird ein Schritt 21 ausgeführt, um eine NULL für einen Quotienten  $Q$  und eine NULL für eine Variable  $SGN$  aus-

zuwählen. Die Variable SGN ist eine Ausgabe des D-Flipflops 4 und umfaßt eine Ein-Bit-Information, die für ein Vorzeichen kennzeichnend ist. Bei Beenden des Schritts 21 führt das Programm die folgenden Schritte in Reaktion auf einen Divisionsbefehl durch. Insbesondere wird festgestellt, ob die Variable SGN eine NULL ist oder nicht (Schritt 22). Wenn die Antwort in Schritt 22 JA ist, wird ein Schritt 23 durchgeführt für das Subtrahieren des Divisors B vom Dividenten A und das Speichern des Ergebnisses als einen Rest C. Genauer gesagt subtrahiert in Schritt 23 der Addierer/Subtrahierer 3 den im B-Register 2 gespeicherten Divisor B von dem im A-Register 1 gespeicherten Dividenten A. Da das Ausgangssignal 62 des D-Flipflops 4 ausnahmslos sofort nach Beginn eines Divisionsbefehls eine NULL ist, wird Schritt 23 notwendigerweise durchgeführt.

Wenn die Antwort in Schritt 22 NEIN ist, d.h. wenn die Variable SGN eine EINS ist, werden der Divident A und der Divisor B addiert (Schritt 24). Genauer gesagt wenn die Ausgabe 62 des D-Flipflops 4 eine EINS ist, addiert der Addierer/Subtrahierer 3 die im A-Register 1 und im B-Register 2 gespeicherten numerischen Daten.

Auf die Schritte 23 bzw. 24 folgt ein Schritt 25 zum Ersetzen der Vorzeichenbit-Daten des Ergebnisses C der Subtraktion oder Addition durch die Variable SGN. Das Vorzeichenbit ist eine NULL, wenn das Ergebnis C positiv ist, und eine EINS, wenn das Ergebnis C negativ ist. Genauer gesagt ist das Vorzeichenbit 9 der Ausgabe des Addierers/Subtrahierers 3, wie es in Schritt 23 oder 24 bestimmt wurde, und das entweder einer NULL oder eine EINS ist, in das D-Flipflop 4 geladen.

Im folgenden Schritt 26 wird das invertierte Vorzeichenbit  $\overline{SGN}$  der Ergebnissdaten, die in Schritt 23 bzw. 24 erzeugt wurden, seriell in das Register 6 geschrieben, während es sequentiell von der LSB-Position verschoben wird. Das invertierte Vorzeichenbit  $\overline{SGN}$  ist eine EINS bzw. eine NULL, wenn das Ergebnis der Operation C positiv bzw. negativ ist. Insbesondere werden die Daten des Vorzeichenbits 9, die vom Addierer/Subtrahierer 3 als Ergebnis der Schritte 23 bzw. 24

ausgegeben wurden, vom Invertierglied 5 invertiert und dann sequentiell in das Register 6 geladen, während sie seriell von der LSB-Position verschoben werden.

In einem auf Schritt 26 folgenden Schritt 27 wird das Ergebnis C aus Schritt 23 bzw. 24 arithmetisch um ein Bit nach links verschoben, und es wird eine NULL in der LSB-Position des A-Registers gespeichert. Der resultierende Wert, d.h. ein Wert der durch das Verdoppeln des Ergebnisses C erzeugt wurde, wird im Register A gespeichert. Genauer gesagt wird die Ausgabe des Addierers/Subtrahierers 3, die für das Ergebnis des Schrittes 23 bzw. 24 kennzeichnend ist, durch die Schiebbeeinrichtung 7 verdoppelt und dann im Register 1 gespeichert.

Die oben beschriebene Abfolge der Schritte 22 bis 27 wird  $N+1$  mal insgesamt wiederholt (Schleife 28). Genauer gesagt wird jedes Mal, wenn die oben beschriebene Abfolge vollständig abgelaufen ist, der Schleifenzähler 8 um 1 (Eins) dekrementiert. Die Schritte 22 bis 27 werden so oft wiederholt, bis der Schleifenzähler 8 den Wert 0 (Null) erreicht. Während eine solche Divisionsverarbeitung ausgeführt wird, verhindert das Steuersignal des Schleifenzählers 8, daß der Programmzähler 11, der in den Mikroprozessor zum Halten der Adresse eines als nächstes auszulesenden Befehls eingebaut ist, aktualisiert wird.

In Fig. 1 kann die gezeigte Schiebbeeinrichtung 7 zwischen das A-Register 1 und den Addierer/Subtrahierer 3 geschaltet werden, wie aus der obigen Beschreibung nachvollzogen werden kann. Der erfindungsgemäße Dividierer kann ausgeführt werden, ohne daß eine spezielle Schaltung erforderlich wäre. Ein Grund dafür ist, daß ein gewöhnlicher Mikroprozessor, insbesondere ein digitaler Signalprozessor, normalerweise folgende Komponenten umfaßt: Register, Schieberegister, eine arithmetische und logische Einheit, ein Flag-Register zum Speichern eines Vorzeichenbits, das zu einem Ergebnis der Rechnung der arithmetischen und logischen Einheit gehört, einen Schleifenzähler und einen Barrelshifter. Folglich wird für den erfindungsgemäßen Dividierer nur eine Minimum an zusätzlicher Hardware benötigt, wie beispielsweise das D-Flipflop 4 aus Fig. 1. Des weiteren genügt ein einziger Maschinenzklus eines Prozessors, damit die Abfolge der Schritte 22 bis 27  $N+1$  mal hintereinander wiederholt wird.

Insgesamt wird nun sichtbar sein, daß die vorliegende Erfindung einen Dividierer liefert, der mit einer hohen Geschwindigkeit arbeitet und einen einfachen Aufbau aufweist. Dieser beispiellose Vorteil kommt von dem einzigartigen Verfahren, bei dem ein Dividend und ein Divisor einer Addition bzw. einer Subtraktion unterzogen werden in Abhängigkeit von einem Vorzeichenbit, das in den Differenzdaten oder Summendaten des Dividenden und Divisors enthalten ist, dann die Vorzeichenbit-Daten invertiert werden und dann die invertierten Vorzeichenbit-Daten seriell in ein Schieberegister eingegeben werden, das einem Quotienten der LSB-Position der Daten zugeordnet ist.

Obwohl die vorliegende Erfindung unter Bezugnahme auf die spezielle veranschaulichende Ausführungsform beschrieben wurde, ist sie nicht auf diese Ausführungsform beschränkt. Es ist zu beachten, daß ein Fachmann die Ausführungsform verändern oder modifizieren kann und dies dabei weiterhin in den Schutzzumfang der vorliegenden Erfindung fällt.

## ANSPRÜCHE

1. Dividierer zum Erzeugen eines Quotienten durch Dividieren eines Dividenden durch einen Divisor mit:

einer ersten Halteeinrichtung (1) zum Halten der den Dividenden kennzeichnenden Dividend-Daten;

einer zweiten Halteeinrichtung (2) zum Halten der den Divisor kennzeichnenden Divisor-Daten; und

einer Operationseinrichtung (3) zum Erzeugen entweder einer Summe oder einer Differenz zwischen den in der ersten Halteeinrichtung (1) gehaltenen Dividend-Daten und den in der zweiten Halteeinrichtung (2) gehaltenen Divisor-Daten; dadurch gekennzeichnet, daß der Dividierer folgendes umfaßt:

eine dritte Halteeinrichtung (4) zum Halten von Vorzeichenbit-Daten, die in Ergebnisdaten enthalten sind, welche für ein von der Operationseinrichtung (3) erzeugtes Operationsergebnis kennzeichnend sind;

eine Invertiereinrichtung (5) zum Invertieren der Vorzeichenbit-Daten, um invertierte Vorzeichenbit-Daten zu erzeugen;

eine erste Schiebeeinrichtung (6) zum sequentiellen Verschieben der invertierten Vorzeichenbit-Daten von einer niedrigstwertigen Bitposition (LSB-Position) immer dann, wenn die invertierten Vorzeichenbit-Daten aus der Invertiereinrichtung (5) eingegeben werden, während die invertierten Vorzeichenbit-Daten gehalten werden;

eine zweite Schiebeeinrichtung (7) zum arithmetischen Verschieben der von der Operationseinrichtung (3) erzeugten Ergebnisdaten um ein Bit nach links, während eine logische Null in einer LSB-Position gespeichert wird, und zum Liefern der resultierenden Daten an die erste Halteeinrichtung (1); und

eine Steuerungseinrichtung (8) zum Steuern der Ausführung der iterativen Divisionsverarbeitung durch die Steuerung der Operationseinrichtung (3), der ersten Schiebeeinrichtung (6) und der zweiten Schiebeeinrichtung (7) derart, daß die Operationseinrichtung (3) die Summe oder Differenz auf der Grundlage der von

der dritten Halteeinrichtung (4) gehaltenen Vorzeichenbit-Daten erzeugt, die zweite Schiebereinrichtung (7) das von der Operationseinrichtung (3) erzeugte Operationsergebnis verdoppelt, die erste Halteeinrichtung (1) das verdoppelte Ergebnis hält, die dritte Halteeinrichtung (4) die Vorzeichenbit-Daten des Operationsergebnisses hält und die Invertiereinrichtung (5) die Vorzeichenbit-Daten invertiert und die invertierten Vorzeichenbit-Daten an die erste Schiebereinrichtung (6) liefert, wobei der Quotient in der ersten Schiebereinrichtung (6) gespeichert wird und wobei im Anfangsstadium der Ausführungssteuerung der iterativen Divisionsverarbeitung die Steuerungseinrichtung (8) die Operationseinrichtung (3) so steuert, daß sie eine Subtraktion durchführt, bei der der Divisor vom Dividenten subtrahiert wird.

2. Dividierer nach Anspruch 1, dadurch gekennzeichnet, daß
  - die erste Halteeinrichtung (1) ein A-Register (1) zum Halten der Divident-Daten enthält;
  - die zweite Halteeinrichtung (2) ein B-Register (2) zum Halten der Divisor-Daten enthält;
  - die Operationseinrichtung (3) einen Addierer/Subtrahierer (3) zum Erzeugen der Summe oder der Differenz zwischen den Divident- und den Divisor-Daten enthält;
  - die dritte Halteeinrichtung (4) ein D-Flipflop (4) zum Halten der Vorzeichenbit-Daten enthält, die in den vom Addierer/Subtrahierer (3) erzeugten Ergebnisdaten enthalten sind;
  - die Invertiereinrichtung (5) ein Invertierglied (5) zum Invertieren der Vorzeichenbit-Daten enthält;
  - die erste Schiebereinrichtung (6) ein Schieberegister (6) zum Halten der vom Invertierglied (5) ausgegebenen invertierten Vorzeichenbit-Daten enthält, während die invertierten Vorzeichenbit-Daten immer dann, wenn die invertierten Vorzeichenbit-Daten in die erste Schiebereinrichtung (6) eingegeben werden, von der LSB-Position sequentiell verschoben werden;
  - die zweite Schiebereinrichtung (7) eine Schiebereinheit (7) zum arithmetischen Verschieben der von der Operationseinrichtung (3) erzeugten Ergebnisdaten um ein Bit nach links enthält, während eine Null in der LSB-Position gespeichert wird,

und zum Liefern der resultierenden Daten an die erste Halteeinrichtung (1); und daß

die Steuereinrichtung (8) einen Schleifenzähler (8) enthält zum Steuern der Operationseinrichtung (3), der ersten Schiebereinrichtung (6) und der zweiten Schiebereinrichtung (7) derart, daß die iterative Divisionsverarbeitung wiederholt ausgeführt wird.

3. Dividierer nach Anspruch 2, dadurch gekennzeichnet, daß das D-Flipflop (4) die Vorzeichenbit-Daten einen Maschinenzklus lang hält, während dessen ein Divisionsbefehl ausgeführt wird.

4. Dividierer nach Anspruch 3, dadurch gekennzeichnet, daß der Addierer/Subtrahierer (3) die Differenz zwischen dem Dividenten und dem Divisor erzeugt, wenn die vom D-Flipflop (4) gehaltenen Vorzeichenbit-Daten Null sind, und eine Summe des Dividenten und des Divisors erzeugt, wenn die Vorzeichenbit-Daten eine (logische) Eins sind.

5. Dividierer nach Anspruch 4, dadurch gekennzeichnet, daß der Schleifenzähler (8) mit der Anzahl an Wiederholungen  $N+1$  geladen wird (wobei  $N$  die Anzahl der Bits ist, die dem Bruchteil des numerischen Formats des Quotienten zugewiesen wird), die eine iterative Sequenz wiederholt werden soll, die aus folgendem besteht:

einem ersten Schritt, in dem bestimmt wird, ob unter den vom Addierer/Subtrahierer (3) erzeugten Ergebnisdaten die Vorzeichenbit-Daten eine Null sind oder nicht;

einem zweiten Schritt, in dem abhängig von den Vorzeichenbit-Daten die Summe oder Differenz zwischen dem im A-Register (1) gespeicherten Dividenten und dem im B-Register (2) gespeicherten Divisor erzeugt wird;

einem dritten Schritt, in dem die Vorzeichenbit-Daten, die in den im zweiten Schritt erzeugten Ergebnisdaten enthalten sind, in das D-Flipflop (4) eingegeben werden;

einem vierten Schritt, in dem die invertierten Vorzeichenbit-Daten der im zweiten Schritt erzeugten Ergebnisdaten seriell von der LSB-Position des Schieberegisters (6) in das Schieberegister (6) eingegeben werden, das einen Wert Q als Quotienten speichern soll; und

einem fünften Schritt, in dem die im zweiten Schritt erzeugten Ergebnisdaten verdoppelt werden und ein verdoppeltes Ergebnis im A-Register (1) gespeichert wird;

wobei der Schleifenzähler (8) jedesmals dekrementiert wird, wenn die Sequenz vollständig durchgeführt ist, und wobei der Schleifenzähler (8) veranlaßt, daß die Sequenz so oft wiederholt wird, bis er selbst Null wird.

6. Dividierer nach Anspruch 5, dadurch gekennzeichnet, daß der Dividierer in einen Microprozessor eingebaut ist, der einen Programmzähler (11) zum Halten einer Adresse eines als nächstes auszulesenden Befehls enthält, wobei der Schleifenzähler (8) an den Programmzähler (11) ein Steuersignal liefert, das das Aktualisieren des Programmzählers (11) so lange verhindert, bis die Sequenz N+1 Mal wiederholt worden ist.



FIG. 1

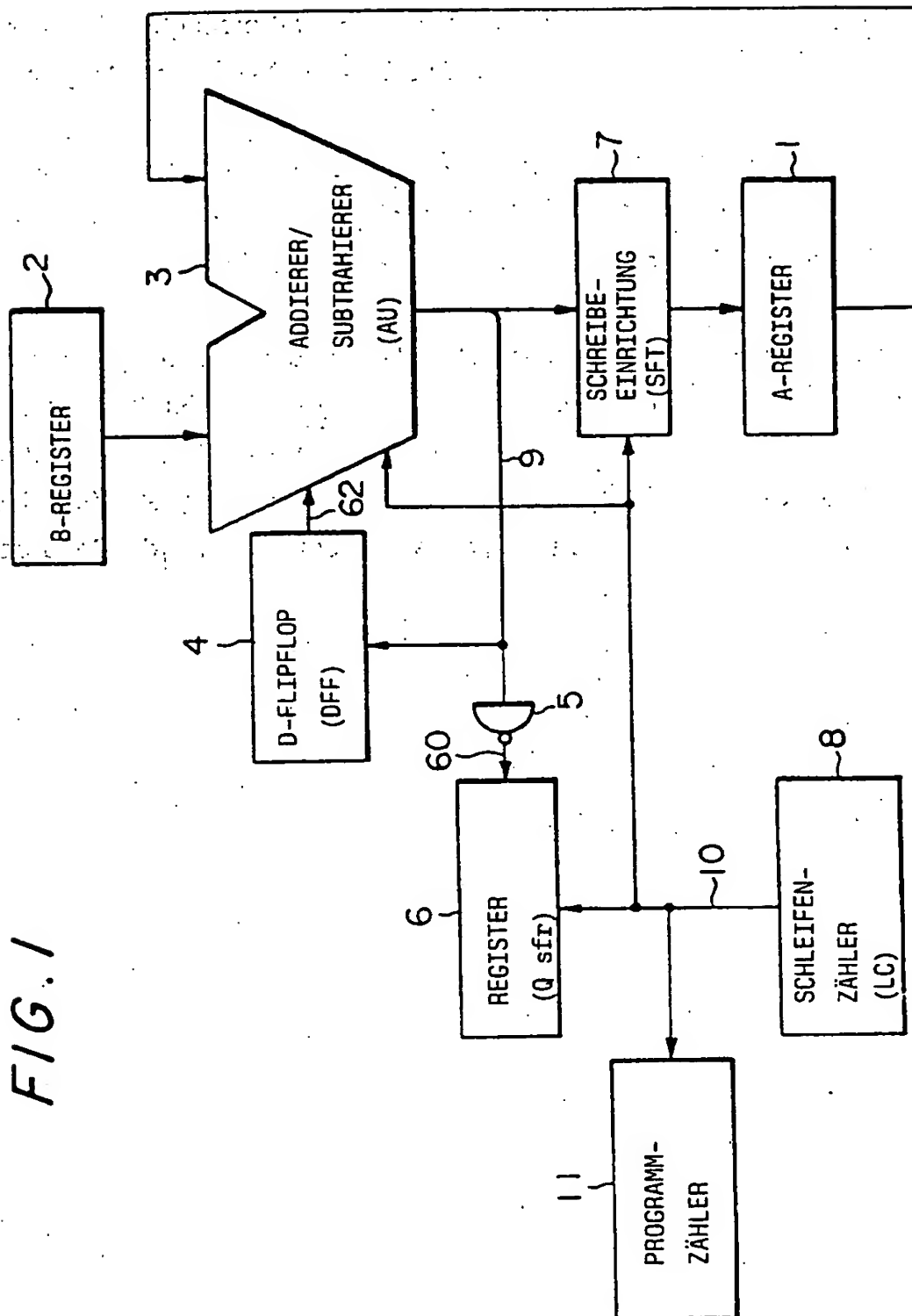


FIG. 2

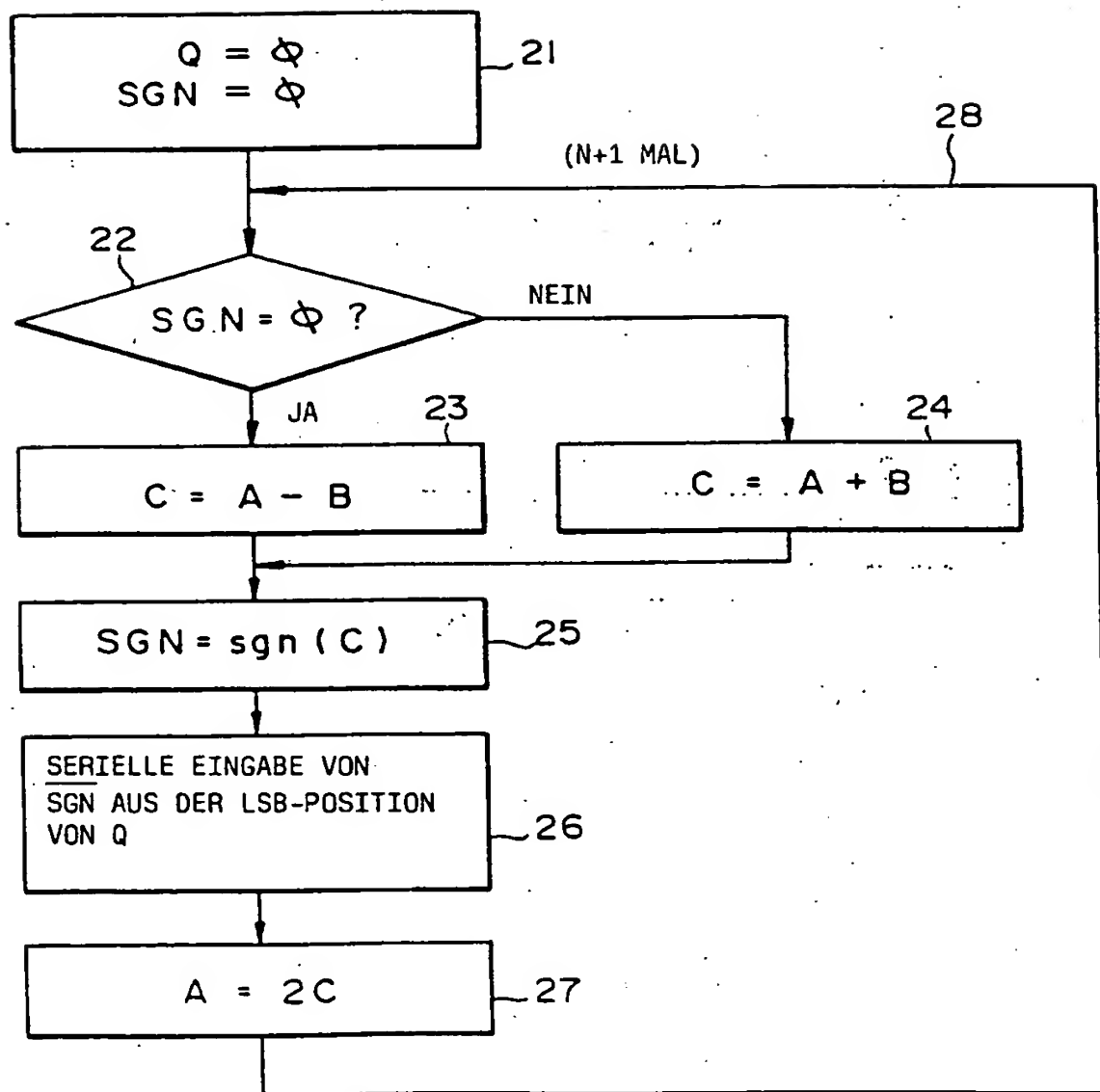


FIG. 3

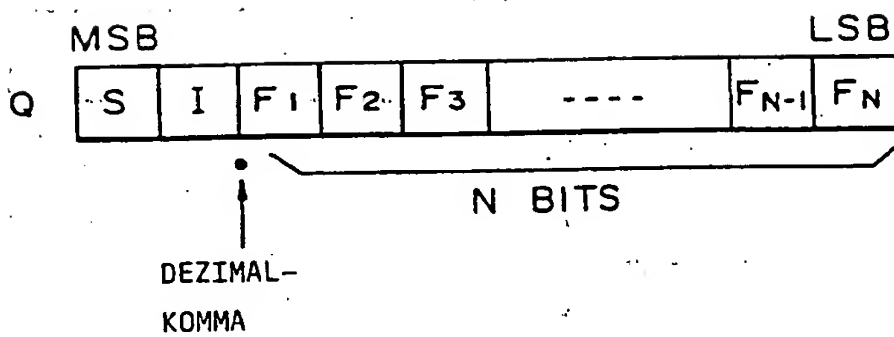


FIG. 4

